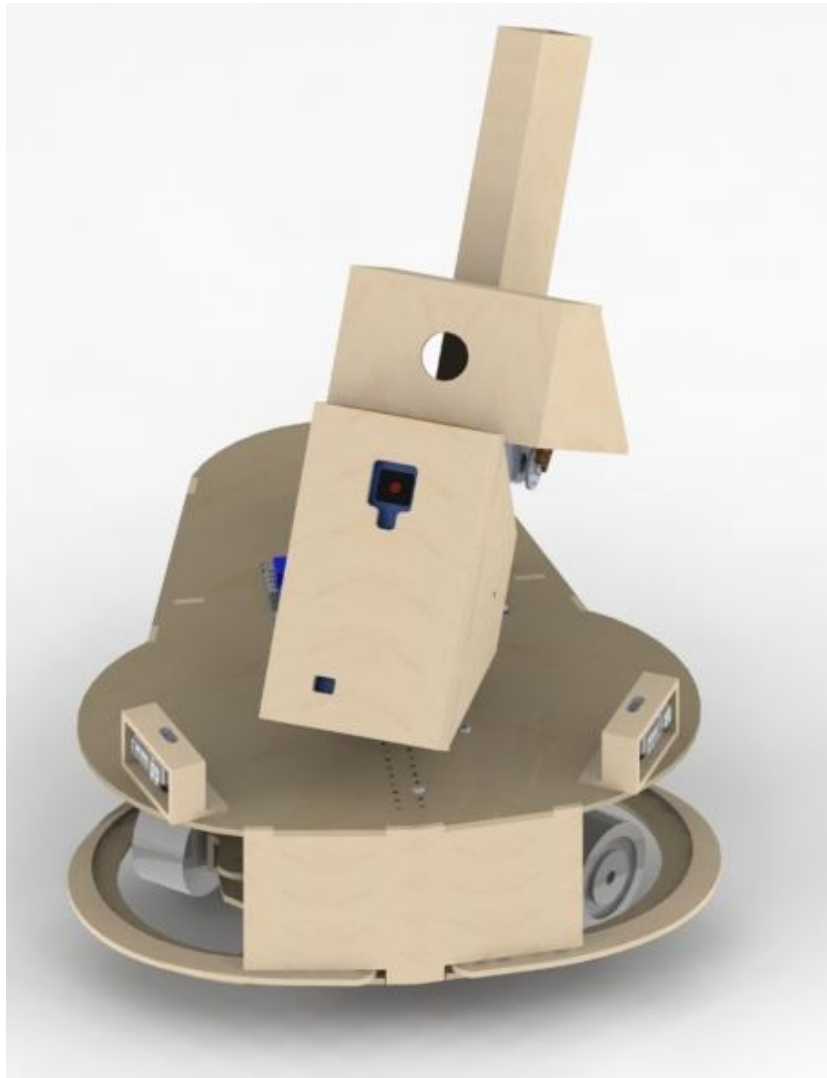


# MinuteMachine

Devin Holland  
April 21, 2014



University of Florida  
Department of Electrical and Computer Engineering  
EEL4665C - IMDL **Spring 2014**  
Instructors: A. Antonio Arroyo, Eric M. Schwartz  
TAs: Joshua Weaver, Andy Gray, Nicholas Cox, Daniel Frank

# Table of Contents

Abstract .....	3
Executive Summary .....	3
Introduction.....	5
Integrated System.....	6
Mobile Platform.....	7
Actuation.....	7
Sensors.....	8
---- Special Sensor.....	8
Behaviors.....	9
Audio to Serial PCB & Results.....	10
Conclusion.....	12
Documentation.....	12
Appendices.....	12

# Abstract

Robotics exist to improve humanity by taking over the repetitive, uncomfortable, and dangerous aspects of our lives. One of the most confrontationally dangerous situations to be in is where the danger comes from humans themselves. I am modeling my robot with the duties of an urban police officer or soldier. I want my robot to be able to provide a protected environment by emulating the duties of law enforcement. By watching out for situations in progress it will be able to interfere and prevent any unnecessary harm to the victim. When the MinuteMachine identifies unregistered weaponry in the form of a concentric color pattern it will lock on to the offending weaponry and fire.

## Executive Summary

MinuteMachine is an autonomous robot created to navigate a typical hallway that you might find in an office environment. The main objective while navigating an area is to autonomously identify concentric colored targets and shoot them. One of the goals for this project was to build a platform that is completely self contained, requiring no external signals for any system. This was achieved by integrating an off the shelf Android device to handle the vision system.

Locomotion is provided by a unique drive system. A single motor in the rear of the robot provides all the power. This single motor is attached to a single wheel that can pivot 180\* to enable turning in place. In addition a pitman arm is used to drive another steering mechanism in the front. MinuteMachine was designed this way to gain the ability to strafe at an angle towards a target.

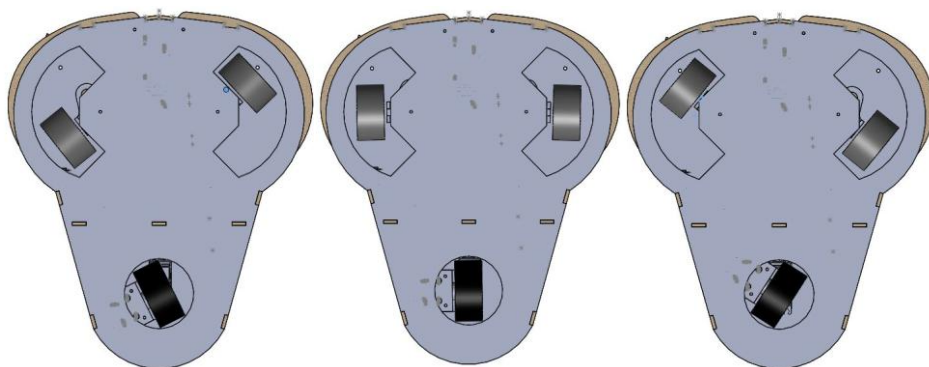


Figure 1: MinuteMachine's dual steering capabilities

Two bump sensors and two IR sensors were placed on the front for obstacle avoidance. Concealed inside the MinuteMachine can pivot the two IR sensors synchronously to maximize the sphere of detection for any obstacles or walls. Inside and out of view resides everything

required to get the MinuteMachine powered and running. Included is the 12v lipo battery, motor controller, 5v regulator and my power and signal distribution PCB, my audio to serial PCB, microcontroller, 3 servos, LCD display and system buttons. On top of the main platform resides a pan & tilt rig containing a Samsung S4 (Android device) and dart launching device

When MinuteMachine is first turned on, the voltage from the 12v lipo battery is split between a motor controller and a 5v regulator that powers the microcontroller and 10 additional devices. The LCD will immediately prompt the user to select a mode from a list after a short bootup sequence. The modes include a 'Full Mode' to launch the main program, a 'Turrent Mode' to isolate the targeting system, and a 'Calibration Mode' to set two custom threshold values for obstacle avoidance.

In 'Full Mode' the MinuteMachine will use the microcontroller to navigate an area, this mode relies on an interrupt that automatically receives coordinates from the android device. When a set of coordinates dependant on the center of mass for both colors passes a series of filters built into the Android itself passes, the data is passed using USART over the audio jack to the microcontroller. When a positive match has been detected, MinuteMachine will stop and look at the specified coordinates and initiate 'Target Mode'. If within the next five seconds another set of coordinates are captured and within the tolerances for the center of the screen a firing sequence begins. The launcher motor will power up and a single dart will be fed into it. MinuteMachine will continue to fire at the target until it disappears and will resume navigating the area.

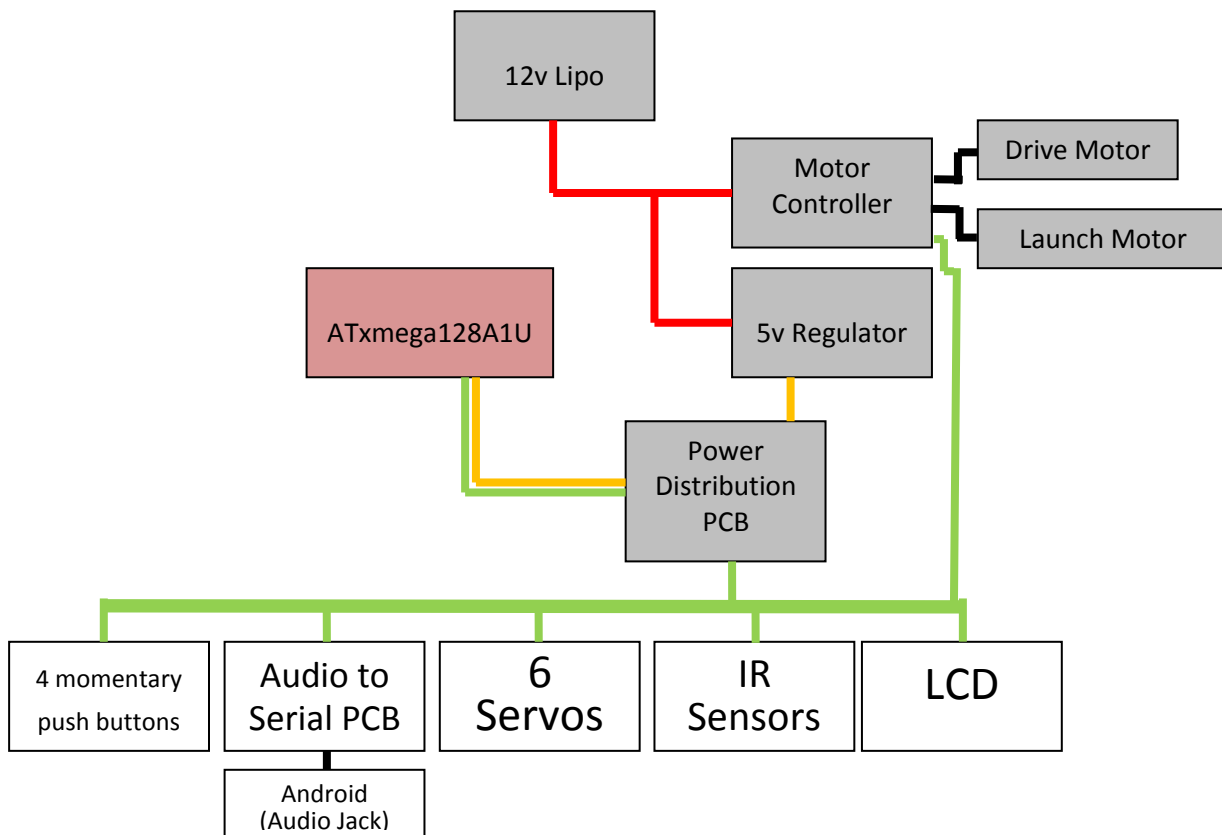
# Introduction

The goal for this robot is to detect a disturbance and neutralize the situation. To detect a disturbance the robot will scan for a certain color pattern to attempt to find unregistered weaponry and shoot at it. Afterwards the robot will resume patrolling. I've always wanted to build robotics that would integrate into our everyday lives and help us in some way. For this project I wanted to do something ambitious. I wanted to learn how to use a sensor that has previously been out of my reach, I wanted a vision system. Also I wanted to recycle old smart phones and use them to their full potential. After learning about a vision library called OpenCV that could be installed on my android I was set. I Learned how to write Android apps and how to incorporate OpenCV into it.

This report will cover the construction and go over some details of the MinuteMachine. The machine will operate in one of two user selected modes: 'Full Mode' and 'Turret Mode'. In Turret Mode the MinuteMachine will remain stationary and scan for any unregistered weaponry. In Full Mode the MinuteMachine will navigate an area and flip to Turret Mode if a target is spotted.

## Integrated System

The final architecture of MinuteMachine can be seen in **Figure-2**. The secondary systems are controlled by an ATxmega128A1U microprocessor. The ATxmega will be connected through USART over the audio jack to an onboard Android mobile that will handle the visual processing. The ATxmega controls all functions with the Android acting like a sensor which only provides data. The block diagram in **Figure-2** below shows all the components and ties between them.



**(Figure-2)**  
MinuteMachine system block diagram

## Mobile Platform

The vehicle of MinuteMachine is a unique reverse powered tricycle. This design was chosen over others because it'll give all the benefits of a steering and tank tread design, combined it will provide a variety of driving capabilities. One main benefit of this design is the ability to strafe towards a target.

Designed to drive over indoor surfaces and among multiple objects, the drive system is housed and protected inside the frame which provided a low profile to roll over tile and smooth carpeting.

## Actuation

MinuteMachine uses a HG16 Copal motor and one Hitec HS-422 servo to power and steer the rear drive system respectively, the single rear wheel will be able to rotate 180° to provide pivoting and minor course corrections. Another Hitec servo controls the front steering which provides major course corrections and interesting maneuvers.

The firing mechanism will be mounted on a pan & tilt system powered by another 2 Hitec servos. The brushed motors use two counter spinning drums that are spun up to their maximum speed. A foam dart is fed in between the two drums using a mini servo. The turret relies on gravity to reload the turret after the mini servo returns to the docked position.

Lastly the sweeping IR sensors are allowed to pivot only when a Hitec servo pulls a string inside the chassis which rotates a large wooden disk. Both the IR sensor and the wooden disk are fixed to a 1.5" 4-40 screw. When returning to the front the disks use a set of springs to pick up the slack in the line.

## Sensors

MinuteMachine's two important senses are infrared distance measuring and an onboard vision system. To visually scan its surroundings the robot will use the onboard camera built into the Android. Then by using OpenCV to detect color blobs it will send filtered data to the microcontroller to decide what to do. A custom app was needed to collect and process the information from the Android and communicate with the microcontroller. The robot will also use IR light to detect the world around it, and if they were to fail bump switches would let the robot know it encountered something solid.

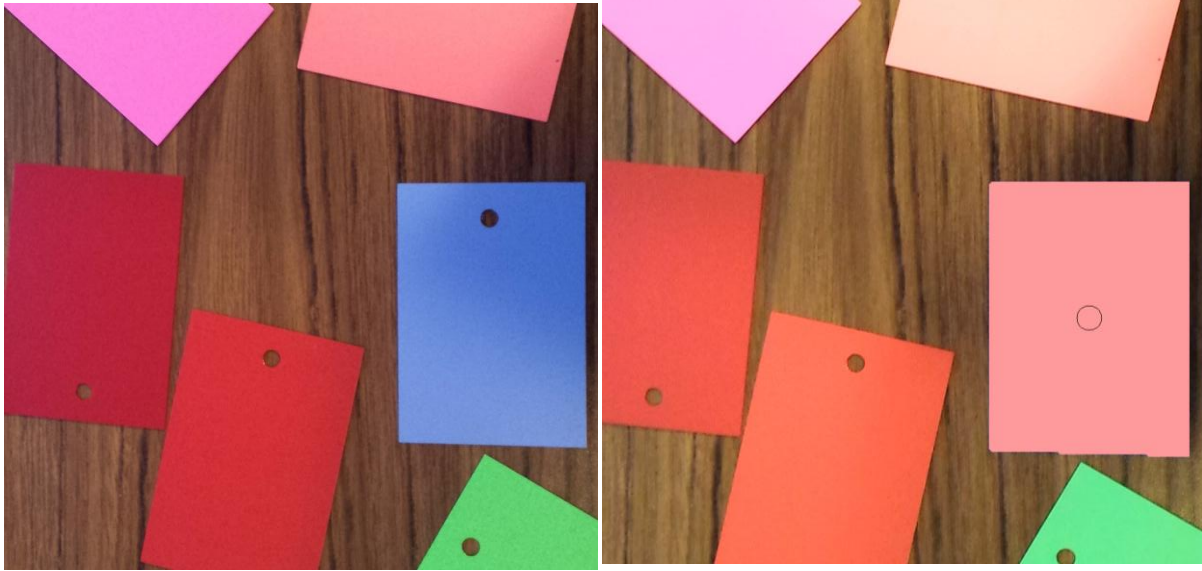
## Special Sensor

The special sensor used on MinuteMachine is a camera which is incorporated in an Android device. The live feed will be analyzed frame after frame by a software library known as OpenCV. This OpenCV library can be used on many different platforms including Android. To use it I have written an app in Eclipse and could possibly distribute it throughout the world in the app store.

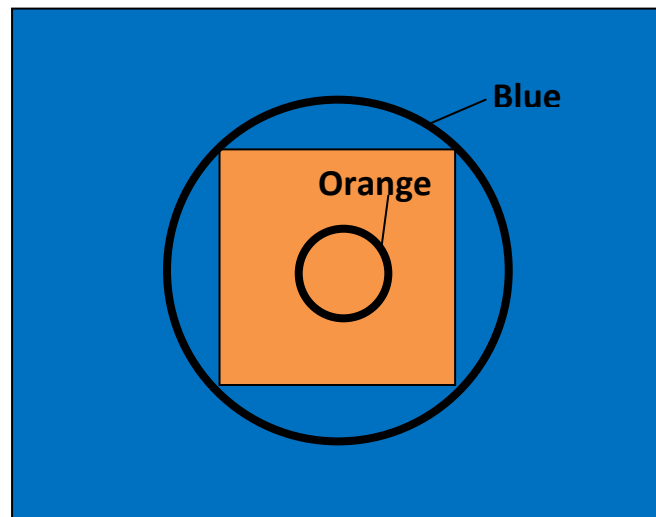
The inspiration to use an onboard camera came from the desire to learn how to use one. Having the skill to work with discarded old smart phones opens a plethora of avenues that I could go down that I couldn't before. The learning curve however is very steep and I have spent most of my time slugging through debugging errors. However I successfully managed to learn everything I needed to see and transmit the sensor data. My app can calculate the X & Y coordinates of the center weight of two separate colors.

When the app first boots it scans its environment for a predetermined set of colors, an example of this can be seen in **Figure-2**. To reduce the number of false positives, the colors will need to follow a pattern, sharing the same center weight as seen in **Figure-2(Bottom)**. Once the colors have been detected the coordinates are sent via USART over the audio jack. The coordinates will be centered on by the pan/tilt system and shot at by an onboard Nerf gun.





**(Figure-2 ) Left: Before OpenCV | Right: After OpenCV)**  
**Bottom: Matching center weights signified by circles**



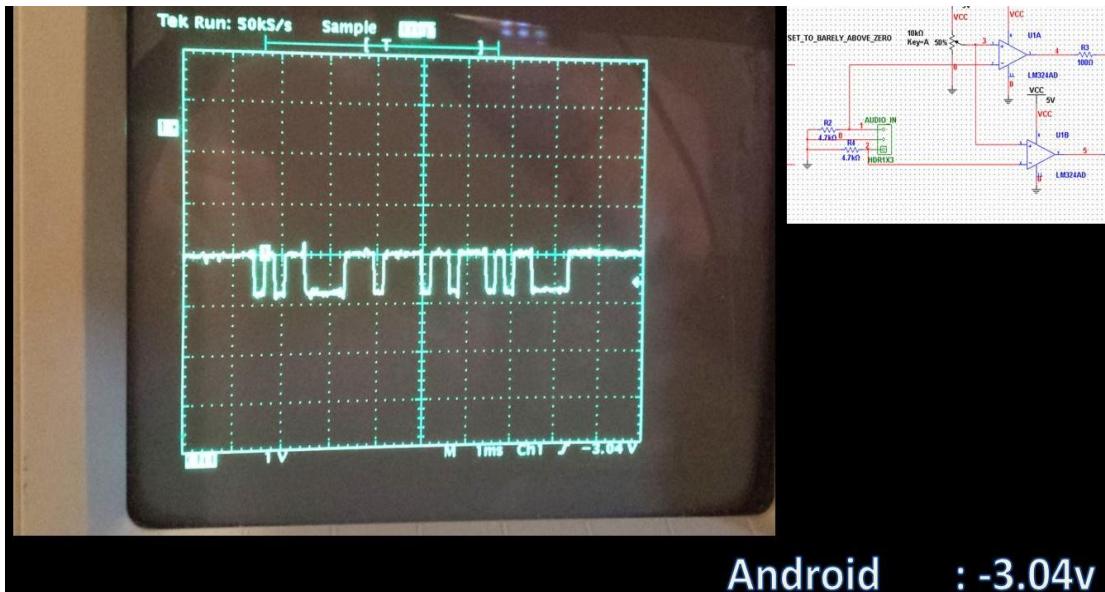
To gather sensor data the Android mobile will be brought to different lighting conditions across campus to see if it can still detect the predicted colors. Both indoor lighting of the New Engineering building, the IMDL lab and my home, followed by general outdoor lighting will be tested.

## Behaviors

MinuteMachine's default behavior is to avoid obstacles while patrolling. In this default behavior it will observe it's surroundings. If at any time it detects a solid color which will represent a weapon it will switch into a second mode that will target the object and fire a dart.

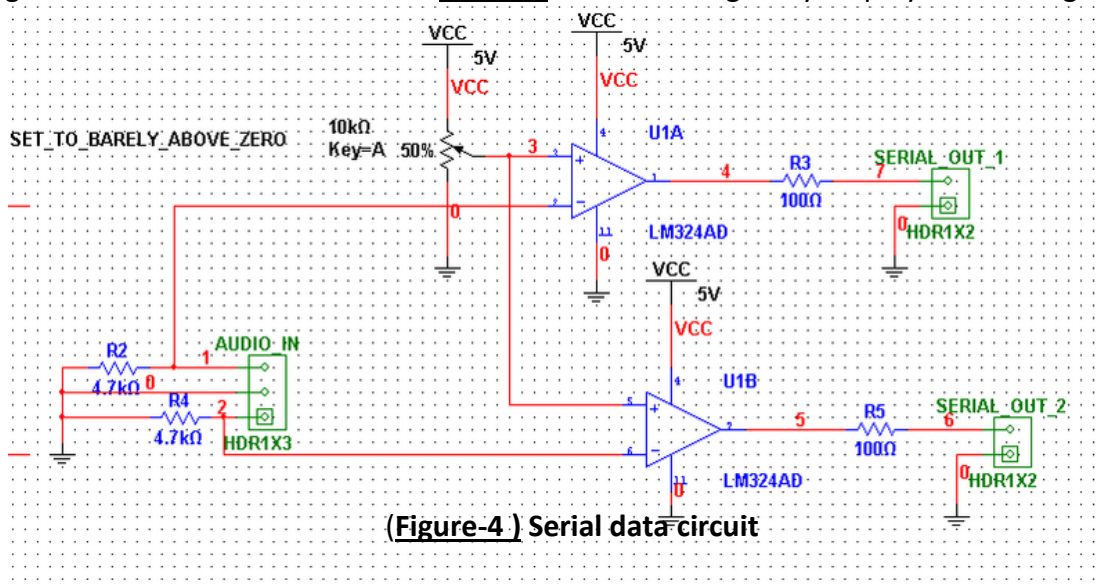
## Audio to serial PCB & Results

One of the most difficult things that I encountered with the android was how to send over the data to the microcontroller. At first a Bluetooth module was going to be used but due to cascading errors and an approaching deadline it was tossed for a more reliable method. Utilizing various online sources I was able to get a program working that would send strings of data over the audio jack using USART protocols. However, the signal straight from the Android was weak and negative as seen in **Figure-3a**.



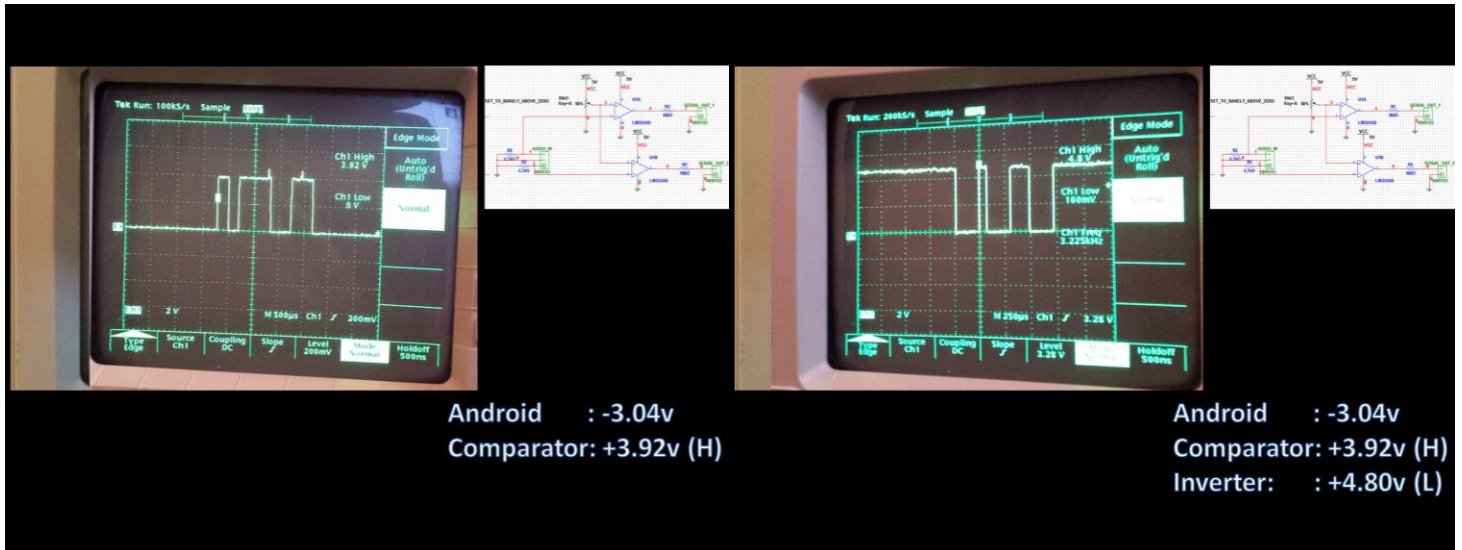
(Figure-3a) Serial data from Android

Using the circuit schematic as seen in **Figure-4** I was able to greatly amplify the data signal

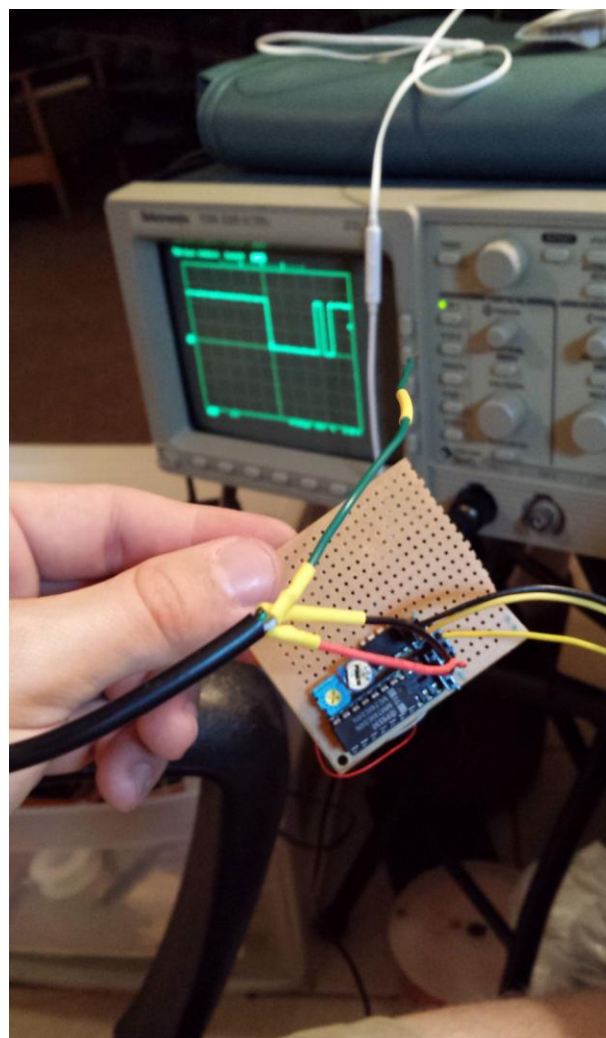


(Figure-4) Serial data circuit

However the signal was active high as seen in **Figure-3b**. So an inverter was added to the output to make the data signal compatible with the microcontroller as seen in **Figure-3c**. With a working circuit I soldered it together and gave it one final test as seen in **Figure-5**.



(**Figure-3b (Left) 3c (Right)** Serial data circuit output  
(**Figure-5 (Below)**) Finalized circuit.



## Conclusion

I have learned a lot from building MinuteMachine. I designed it so I could always come back and repurpose it. Taking advantage of the T-Tek to design all the parts I needed in a computer was a fantastic tool that I never had access to. The ability to design anything I could come up with was invaluable. Later on in the semester I tried to make some simple parts by hand and even my best work was shadowed by the same time spent on a computer. With all the experience I got from working at the IMDL, my next design will be more maintenance friendly with less snaking wires because I now know how to plan ahead.

Before the IMDL, I had never worked with Androids or OpenCV. In order for me to finish my robot I had to learn how to program in both of their native languages; Java and C++. This was a major accomplishment for me when I got everything to cooperate a few weeks before the end of the semester. I feel as if I have only scratched the surface of what I can do with OpenCV and will continue to upgrade the software of MinuteMachine.

## Documentation

- ATxmega128A1U datasheet: [http://www.atmel.com/Images/Atmel-8331-8-and-16-bit-AVR-Microcontroller-XMEGA-AU\\_Manual.pdf](http://www.atmel.com/Images/Atmel-8331-8-and-16-bit-AVR-Microcontroller-XMEGA-AU_Manual.pdf)
- Android Serial over Audio resource: [http://robots-everywhere.com/re\\_wiki/index.php?title=Serial\\_on\\_Android\\_using\\_the\\_audio\\_port](http://robots-everywhere.com/re_wiki/index.php?title=Serial_on_Android_using_the_audio_port)
- ANT 100 Motor controller: <http://www.robotmarketplace.com/products/LB-ANT100-2.html>
- Samsung S4 User Manual: [http://cache.vzw.com/multimedia/mim/sam\\_galaxy\\_s4/samsung\\_galaxy\\_s\\_4\\_um.pdf](http://cache.vzw.com/multimedia/mim/sam_galaxy_s4/samsung_galaxy_s_4_um.pdf)
- OpenCV Resource: <http://opencv.org/>
- Sharp IR Rangefinders: [http://www.sharpsma.com/webfm\\_send/1489](http://www.sharpsma.com/webfm_send/1489)

## Appendices

MinuteMachine is open-source and everything can be found at:

<https://sites.google.com/site/MinuteMachineIMDL/>